A Location-free Prediction-based Sleep Scheduling Protocol for Object Tracking in Sensor Networks

Jue Hong*, Jiannong Cao†, Yingpei Zeng*, Sanglu Lu*, Daoxu Chen* and Zhuo Li*
*State Key Laboratory for Novel Software Technology, Nanjing Univ., Nanjing, China
†Department of Computing, Hong Kong Polytechnic Univ., Hung Hom, Kowloon, Hong Kong, China

***Abstract*—Sleep scheduling protocols are widely used in wireless sensor networks for saving energy in sensor nodes. However, without considering the special requirements of object tracking, conventional sleep scheduling protocols may lead to intolerable degradation of tracking qualities when they are used in object tracking applications. To handle this problem, sleep scheduling protocols tailed for object tracking have been proposed recently. For saving energy while maintaining satisfactory tracking qualities, these protocols proactively awaken sensors according to the prediction of objects' movement. Such sleep scheduling protocols are called the prediction-based sleep scheduling protocols. Most existing prediction-based sleep scheduling protocols require sensor nodes to know the locations of themselves, which may not always be available. In this paper we propose a Location-free Prediction-based Sleep Scheduling protocol (LPSS) for object tracking in sensor networks. LPSS guarantees the coverage level, an important tracking quality in most applications, which is defined as the number of sensors simultaneously detecting the object. In LPSS, when a sensor detects the object, it will emit a signal, namely the sensing stimulus. Sensors decide to wake up or not based on only the received sensing stimulus, the prediction models and the required coverage level, without the requirement of location information. We implement LPSS with two most popular prediction models: the Circle-based and the Probability-based prediction models. Experiment results show that LPSS not only provides qualified coverage levels, but also saves about 40% to 70% energy compared with existing location-free protocols. Moreover, the energy cost of LPSS is close to the ideal approach using accurate location information in terms of the number of awakened nodes.**

I. Introduction

Object tracking is a typical application of wireless sensor networks. During the tracking process, mobile objects should be observed with special requirements on tracking qualities, e.g. detection latency, coverage level and so on [1]–[4]. In a densely deployed sensor network, these requirements can be guaranteed by keeping all sensors active. However, the scarce and unrenewable energy resource on each sensor will be depleted quickly. Therefore, the sleep scheduling technique has been widely used, which helps decrease the number of simultaneous awakened sensors by switching them on and off.

However, without considering the special requirements of object tracking, conventional sleep scheduling protocols may lead to intolerable degradation of tracking qualities, e.g. long detection latency and high tracking error [4], when they are used in object tracking applications. To save energy while maintaining satisfactory tracking qualities, sleep scheduling protocols for object tracking have been proposed recently [5]–[9]. In these protocols, sensors are proactively awakened according to the prediction of objects' movement. Only sensors around the predicted locations of objects will be awakened to detect the object, while others remain sleep. As a result, energy can be greatly conserved without the deterioration of tracking qualities. Such sleep scheduling protocols are called the prediction-based sleep scheduling protocols.

Based on different prediction models, existing prediction-based sleep scheduling protocols can be classified into three categories: circle-based (e.g. [2], [6]), kinematics-based (e.g. [5], [7]) and probability-based (e.g. [9], [10]). Most existing protocols require sensors to know their locations in order to locate the moving object. Moreover, in kinematics-based and probability-based prediction protocols, sensors need to sense the velocities of objects. However, in many situations, sensors do not necessarily have hardware or software for location, nor velocity-detection equipments (e.g., see [2], [11]–[15]). Therefore, without the location information of sensors, existing protocols cannot be used.

In this paper, we propose LPSS, a Location-free Prediction-based Sleep Scheduling protocol for object tracking in sensor networks. LPSS is the only predication-based sleep scheduling protocol that does not require sensors to know their location information except one existing protocol [2]. In LPSS, whenever a sensor detects the object, it will emit a signal, namely the sensing stimulus. The sensing stimulus can be implemented as physical signals like ultrasound, RF impulse, or even a simple message indicating the signal strength. Sensing stimulus carries the implicit information of the tracked object, and will cause the reaction of other sensors. Based on the received sensing stimulus, the prediction models and the requirements of tracking qualities, sensors can decide when to wake up to detect the object.

Because the kinematic rules depend on objects' categories and differ from each other, no unified kinematics-based prediction model exists so far. Thus the kinematics-based sleep scheduling protocol is seldom used. So, in this paper, we present the design of LPSS with the circle-based and the probability-based prediction models. We focus on the tracking quality of coverage level, which is defined as the number of sensors simultaneously detecting the object. The coverage level is an important tracking quality in most applications, directly affecting the tracking error [4] and the resolution [13]. The savings in energy cost is measured in terms of the number of simultaneously awakened nodes during the tracking process. Experiments show that LPSS approximates closely to

978-1-4244-4634-6/09/$25.00 ©2009 IEEE

the energy cost of the ideal approach with accurate location information, and saves 40% to 70% energy compared to existing location-free protocols. Besides, the required coverage level is well guaranteed by LPSS.

The rest of this paper is organized as follows. We briefly review the existing work in Section II. We present the pertinent models and assumptions in Section III. We give an overview of LPSS in Section IV and then the detailed design in Section V. We address the implementation issues in Section VI, and evaluate the performance of LPSS in Section VII. We conclude this paper in Section VIII.

II. Related Work

Object tracking in wireless sensor networks attracted a lot of attention in the last decade (e.g. [1], [2], [4], [11]–[18]). Although sleep scheduling has been well studied as an energy saving technique [19]–[24], existing sleep scheduling protocols did not consider the requirements of object tracking applications and thus may lead to degradation of tracking qualities. To provide a better tradeoff between energy efficiency and tracking qualities, sleep scheduling protocols for object tracking have been proposed recently [5]–[9]. These protocols, called prediction-based sleep scheduling protocols, awaken sensors based on the prediction of objects' movements. Based on different prediction models, existing prediction-based sleep scheduling protocols can be classified as circle-based, kinematics-based and probability-based.

The *circle-based* model is the simplest and most commonly-used prediction model. Given the current location of the object, the predicted locations are within a radius determined by the maximum velocity of the object. Gui et al. used this model to predict the locations of objects and proposed the Proactive Wakeup (PW) scheme [2]. Wang et al. also used the circle-based model in their dynamic energy management approach [6]. Because the objects' movement are usually restricted by some physical laws in real world, the *kinematics-based* model is proposed. Jeong et al. considered the kinematics of vehicles and proposed a Minimal Contour Tracking Algorithm to reduce the number of awakened nodes in [7]. Xu et al. proposed a Prediction-based Energy Saving scheme based on three heuristics of the kinematics of objects to reduce the missing rate and save energy [5]. However, the kinematic rules depend on objects' categories and differ from each other, and thus no unified kinematics-based prediction model exists so far. Hence the kinematics-based approaches are seldom used. The *probability-based* model is proposed for objects whose motion patterns follow some given distributions. The Gaussian distribution is used most frequently to model objects' motion. Jiang et al. proposed a Target Direction-based Sleep Scheduling for objects following the linear and the Gaussian distributions in [9]. Seshadri et al. also used the Gaussian distribution to model objects' motion and predicted their locations [10]. Besides, Fuemmeler et al. modeled objects' motion with the Markov process and proposed the sleep polices in [8].

Most above prediction-based sleep scheduling protocols require sensors to know their locations and thus cannot be used in scenarios without location information. In particular, the PW scheme in [2] awakened sensors base only on communication and can be regarded as location-free. However, without a deliberate consideration of object motion, PW has a high energy cost.

III. Preliminaries

In this section we present models and assumptions. We denote a circle centering at p with radius r by $C(p, r)$, and the distance between locations (or points) u and v by $\|uv\|$.

A. Network and Sensing Model

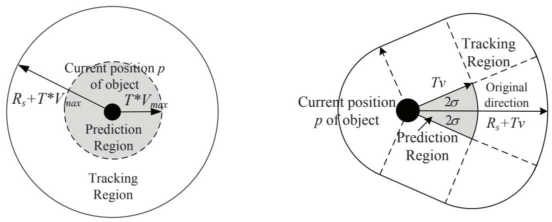
We assume that n sensor nodes are uniformly deployed in a two-dimension plane with a high density ρ. The communication range of each node is a circle with radius R_c. Two nodes can hear each other iff the distance between them is less than or equal to R_c. We also assume that R_c is large enough to make the network connected.

The sensing range of each sensor is also assumed to be a circle with radius R_s. We follow a very common sensing model that each sensor knows only the appearance of objects in its sensing range, and cannot sense other information like velocity or location of objects (e.g. [2], [11]–[15]). Besides, sensors are incapable to obtain their locations. This model describes the simplest and most generalized scenarios of sensor networks.

The running of each sensor is divided into rounds with length T. Each round begins with a sensing phase of a long period $T-\tau$ ($\tau > 0$), and ends with a communication phase of a short period τ. At the beginning of each round, each sensor wakes up according to the computed *wake-up probability*. In the sensing phase of each round, the communication modules of sensors are powered off. The sensing module of a sensor will be turned on to detect the object if it is awakened, otherwise will be turned off to save energy. In the communication phase of each round, sensors turn on their communication modules for collaboration and data delivery. We assume that each node is synchronized with its neighbors and the MAC protocol can guarantee the successful sending and receiving of messages.

B. Motion Pattern of Object

We focus on the problem of tracking a single object. The maximal speed of the object is known as V_{max}. We assume the object follows two most common motion patterns, namely the *random motion pattern* and the *Gaussian motion pattern* [9], [10]. In the random motion pattern, the object changes its speed within [0, V_{max}] and changes its moving direction arbitrarily. In the Gaussian motion pattern, object moves with a constant speed $v \in (0, V_{max}]$, and changes its moving direction only at the beginning of each round according to the zero-mean Gaussian distribution with $\sigma \in [0, \frac{\pi}{6}]$. The assumption of $\sigma \in [0, \frac{\pi}{6}]$ means that the object will not change its direction sharply, which is a common situation of most objects' motion. We do not consider the situations that the object moves to the boundary of the network so that few sensor



(a) Circle-based model (b) Probability-based model

Fig. 1. Prediction and tracking regions with circle-based and probability-based model.

can detect it. According to the object's maximum velocity, and combining factors of R_c, R_s, we let $T = \frac{R_c - R_s}{V_{max}}$ for simplifying the design and analysis of our protocol.

C. Prediction Models

We denote the region containing the predicted locations of object by the *prediction region*, and the region containing the sensors covering the prediction region by the *tracking region*. Here we briefly introduce the circle-based and the probability-based prediction models and determine their prediction and tracking regions.

In the circle-based model, as shown in Fig.1(a), given the current location p of the object, the location of the object after a round T will fall inside the circle $C(p, V_{max}T)$, which is exactly the prediction region. Since an object with location p' can be detected by only the sensors inside $C(p', R_s)$, the tracking region is $C(p, V_{max}T + R_s) = C(p, R_c)$.

We then present the probability-based model for objects following the zero-mean Gaussian motion pattern. We denote the change of moving direction by α. According to the properties of Gaussian distribution $\alpha \sim N(0, \sigma^2)$, we have $Pr[\alpha \leqslant |2\sigma|] > 95.44\%$. Therefore, as shown in Fig.1(b), given the current location p and the moving direction θ of the object, the circular sector centering at p with radius TV_{max} and central angle 4σ ($[\theta - 2\sigma, \theta + 2\sigma]$), can be employed as the prediction region. The corresponding tracking region is the irregular shape in Fig.1(b).

D. Requirement of Tracking Quality

We consider an important metric of tracking quality in this paper, namely the coverage level, which is the number of sensors detecting the object simultaneously. As in most applications (e.g. [2], [22], [25]), we define the requirement of coverage level as: *the object should be covered by at least k sensors during the whole tracking period with a probability of P_{req}*. Parameter k is called the required coverage level, and P_{req} is called the required coverage probability.

According to aforementioned models and assumptions, to conserve energy and provide satisfactory coverage level, only the sensors inside the tracking regions need to wake up with a probability P_t, which can be computed as follows. Given the location p of the object, only awakened sensors inside $C(p, R_s)$ can detect the object. Let random variable x_i denote the state of the ith node inside $C(p, R_s)$. If the ith node is awakened then $x_i = 1$ otherwise $x_i = 0$. It is apparent that random variables x_is are i.i.d and $Pr[x_i = 1] = P_t$ and $Pr[x_i = 0] = 1 - P_t$. Let $X = \sum_{x_i \in C_d} x_i$, X follows the Binomial distribution and hence

$$Pr[X > k] = \sum_{i=k+1}^{n} \binom{n}{i} P_t^i (1 - P_t)^{n-i},$$

where $n = \lfloor \pi R_s^2 \rho \rfloor$. By settling $Pr[X > k] = P_{req}$, we can compute the qualified P_t.

We conclude this section with the following useful fact.

Fact 1. *Given $C(p_1, r_1)$ and $C(p_2, r_2)$, let $d = \|p_1 p_2\|$, when $r_2 - r_1 \leqslant d \leqslant r_2 + r_1$ (assuming $r_2 > r_1$), the area of $C(p_1, r_2) \cap C(p_2, r_2)$ is:*

$$\mathcal{A}(r_1, r_2, d) = r_1^2 \arccos(\frac{d^2 + r_1^2 - r_2^2}{2dr_1})$$
$$+ r_2^2 \arccos(\frac{d^2 + r_2^2 - r_1^2}{2dr_2}) - \frac{1}{2}((-d + r_1 + r_2)$$
$$(d + r_1 - r_2)(d - r_1 + r_2)(d + r_1 + r_2))^{\frac{1}{2}}.$$

Theorem 1. *If the values of r_1 and r_2 are fixed, when $r_2 - r_1 \leqslant d \leqslant r_2 + r_1$ (assuming $r_2 > r_1$), $\mathcal{A}(r_1, r_2, d)$ monotonically decreases with d.*

Proof. When $r_2 - r_1 \leqslant d \leqslant r_2 + r_1$ and both r_1, r_2 are fixed, $\mathcal{A}'(r_1, r_2, d) = -\frac{\sqrt{-(d-r_1-r_2)(d+r_1-r_2)(d-r_1+r_2)(d+r_1+r_2)}}{d} < 0$. This theorem holds immediately. □

IV. An Overview of LPSS

In this section we briefly introduce the components and the running of LPSS. In LPSS, each sensor switches between two states, namely the *detecting state* and the *tracking state*. The detecting state indicates that sensors are not in the tracking region and thus can wake up with a low probability P_d to conserve energy. On the other hand, the tracking state indicates that sensors are in the tracking region and thus have a high wake-up probability P_t to serve tracking requirements. The transition between the two states of each sensor is controlled by the *stimulus management policy* and the *state transition rule*, which are the core components of LPSS. The stimulus management policy consists of the production, storage and update of the sensing stimulus. The state transition rule specifies the tracking conditions, which control the transition between the detecting and the tracking states of each sensor. The stimulus management policy and the state transition rule together enable the sensors in specific tracking regions to enter the tracking state to guarantee required coverage level. Here,

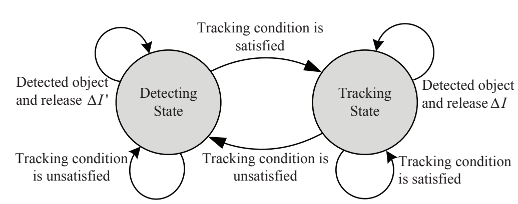


Fig. 2. Transition between the detecting state and the tracking state.

the sensing stimulus are simulated by messages with numbers indicating stimulus strength.

Fig.2 demonstrates the running of LPSS. Each sensor stays in the detecting state initially. When the object first appears in the monitoring area, it will be detected by some sensors in detecting state. Because there are insufficient awakened sensors to carry out quality tracking, these sensors then produce sensing stimulus with a high strength $\Delta_{I'}$ to compel more sensors to switch to the tracking state quickly. For sensors in tracking state, when detect the object they will produce sensing stimulus with a lower strength Δ_I, because there have been already sufficient awakened sensors in expectation. At the beginning of each round, sensors in both the detecting state and the tracking state will check whether the tracking condition is satisfied based on received sensing stimulus to decide to enter the tracking state or the detecting state.

Now we discuss how to determine the probabilities P_d and P_t, as well as the settings of Δ_I and $\Delta_{I'}$. Without the appearance of objects, the wake-up probability P_d in the detecting state is determined by the required detection qualities like detection latency [3], exposure [22] etc., and thus is out of the scope of this paper. Meanwhile, the wake-up probability P_t in the tracking state is determined by the network density ρ, sensing range of sensors R_s and the requirement of coverage level. With the computation of P_t in Section III-D, the required coverage level can be guaranteed by turning only the sensors inside the tracking region into the tracking state. The settings of Δ_I and $\Delta_{I'}$ depend on both P_t and P_d. Since P_d is not discussed here, we omit the details and only present the results used in this paper, i.e., Δ_I is set to an arbitrary positive constant and $\Delta_{I'} = 10\Delta_I$.

In the next section, we present the detailed design of LPSS with the circle-based and the probability-based prediction models. The design includes the stimulus management policies and the state transition rules, which are derived according to the specific shapes of the tracking regions in the two models.

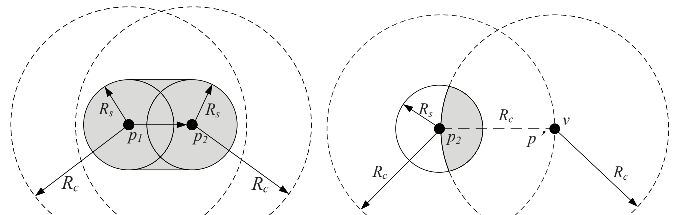
V. LPSS WITH CIRCLE-BASED AND PROBABILITY-BASED PREDICTION MODELS

In this section, we present the design of LPSS with circle-based model first, and then the design of LPSS with probability-based model. In the following description, we assume that the object has appeared and the sensors around it have been in the tracking state.

A. With Circle-based Model

We assume that during an arbitrary round, e.g. the cth round, the object moves from the location p_1 to p_2. In the circle-based model, the tracking region in the $(c+1)$th round is $C(p_2, R_s + TV_{max}) = C(p_2, R_c)$. The stimulus management policy and the state transition rule should be able to make only the sensors inside $C(p_2, R_c)$ enter the tracking state in the $(c+1)$th round.

1) Stimulus Management Policy: As shown in Fig.3(a), the awakened sensors in the gray zone can detect the object in the cth round. To 'extract' the location information of the object,



(a) (b)

Fig. 3. Analysis for LPSS with circle-based model.

we expect that only the awakened sensors inside $C(p_2, R_s)$ produce the sensing stimulus in the communication phase of the cth round. Hence, we let only the awakened sensors detecting the object during a short period $[cT - \epsilon, cT)(\epsilon > 0)$ produce the sensing stimulus. The value of ϵ controls the shape of zone in which the sensors producing the sensing stimulus locate. A smaller ϵ provides a more accurate approximation of $C(p_2, R_s)$. As a result, given the communication radius R_c, all sensors inside $C(p_2, R_s + R_c)$ can hear the sensing stimulus, and a sensor v with location p' can receive the sensing stimulus produced by only the awakened sensors inside $C(p_2, R_s) \cap C(p', R_c)$ (Fig.3(b)). Each sensor accumulates and stores the received sensing stimulus as I. The stored stimulus will be cleared in the next sensing phase.

2) State Transition Rule: Next we design the state transition rule to identify sensors inside $C(p_2, R_c)$. Let $S_{p'}$ denote the area of $C(p_2, R_s) \cap C(p', R_c)$. Given the node density ρ and the wake-up probability P_t, $S_{p'}$ affects the number of awakened nodes and thus the value of I directly. We demonstrate a theorem about $S_{p'}$ first, and then give the transition rule.

Theorem 2. *For each sensor with location p', $p' \in C(p_2, R_c)$ iff $\pi R_s^2 \geqslant S_{p'} \geqslant \mathcal{A}(R_s, R_c, R_c)$.*

Proof. For each sensor with location p', it is apparent that $p' \in C(p_2, R_c) \iff 0 < \|p'p_2\| \leqslant R_c$.

When $R_c - R_s < \|p'p_2\| \leqslant R_c$, according to Fact 1, $S_{p'} = \mathcal{A}(R_s, R_c, \|p'p_2\|)$. According to Theorem 1,

$$R_c - R_s < \|p'p_2\| \leqslant R_c \iff \pi R_s^2 > S_{p'} \geqslant \mathcal{A}(R_s, R_c, R_c).$$

When $0 \leqslant \|p'p_2\| \leqslant R_c - R_s$,

$$0 \leqslant \|p'p_2\| \leqslant R_c - R_s \iff S_{p'} = \pi R_s^2.$$

By summarizing the above two cases this theorem holds. □

Let N denote the number of awakened nodes in the zone with area $\mathcal{A}(R_s, R_c, R_c)$. According to Theorem 2, for a sensor u with location $p' \in C(p_2, R_c)$, the number of awakened sensors inside $C(p_2, R_s) \cap C(p', R_c)$ is supposed to be larger than or equal to N because $S_{p'} \geqslant \mathcal{A}(R_s, R_c, R_c)$. Thus $N\Delta_I$ is supposed to be a lower bound of the accumulated stimulus received by u. Accordingly we propose the tracking condition as $I \geqslant N\Delta_I$ and the state transition rule as follows.

Rule 1. *For each sensor with $I \geqslant N\Delta_I$, enter the tracking state in the next round.*

N is a random variable and thus cannot be computed accurately. Because N serves as a lower bound, simple estimation of N with its expected value will deteriorate the accuracy of Rule 1, since in many cases the actual value of N is below its expected value. Instead, we estimate N with one-side confidence interval $[\widehat{N}, +\infty)$. The details of interval estimation and the corresponding confidence P_1 will be discussed in Section VI. Substituting $\widehat{N}$ for N in Rule 1 and combining the stimulus management policy, we propose LPSS with circle-based model as Fig.4.

For each sensor:
In the sensing phase, e.g. $((c-1)T,\ cT-\tau]$:
1. Compute $[\widehat{N}, +\infty)$ with confidence P_1;
2. If $I \geqslant \widehat{N}\Delta_I$, enter the tracking state, otherwise enter the detecting state;
3. Clear I;

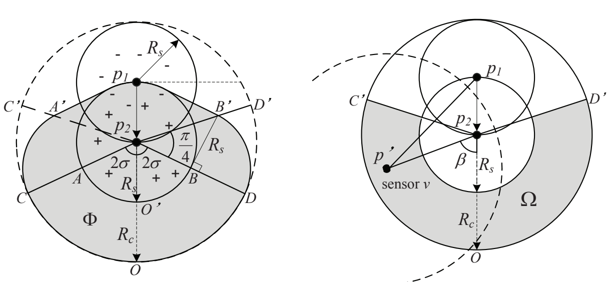
In the communication phase, e.g. $(cT-\tau,\ cT]$:
Concurrently execute the following two steps:
1. Receive sensing stimulus and store in I;
2. If detected the object during $[cT-\epsilon, cT)$: If it is in the detecting state then produce sensing stimulus $\Delta_{I'}$, otherwise produce sensing stimulus Δ_I.

Fig. 4. LPSS with Circle-based Model.

B. With Probability-based Model

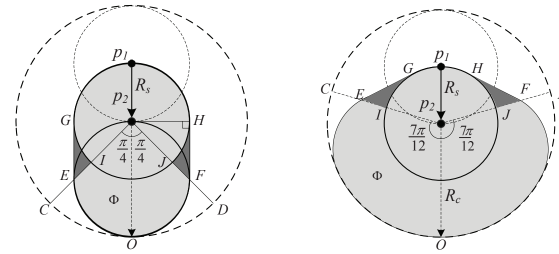
We assume that during an arbitrary round, e.g. the cth round, the object moves from location p_1 to p_2 with constant velocity v. As illustrated in Fig.5(a), in the probability-based model, the prediction region and the tracking region in the $(c+1)$th round are the circular sector p_2AB and the gray zone Φ respectively. Without the knowledge of objects' moving directions and sensors' location information, it is difficult to identify such an irregular tracking region. We make an approximation for the tracking region first, and then derive the stimulus management policy and the state transition rule under the condition of $R_c \leqslant 2R_s$.

1) Approximation of Tracking Region: We simply approximate Φ with region Φ', the union of $C(p_2, R_s)$ and circular sector $p_2C'OD'$, where $\angle D'p_2O = \angle C'p_2O = 2\sigma + \frac{\pi}{4}$ as shown in Fig.5(a). p_2C' and p_2D' intersect Φ at A' and B'



(a) (b)

Fig. 5. Analysis for LPSS with probability-based model.



(a) Case of $\sigma = 0$ (b) Case of $\sigma = \frac{\pi}{6}$

Fig. 6. Proof of Theorem 3.

respectively, where $\|B'B\| = \|A'A\| = R_s$. The following theorem shows that Φ' provides a good approximation of Φ.

Theorem 3. *Φ' covers at least 93.1% the area of Φ.*

Proof. (*Sketch*) Take Fig.6 for example. When $\sigma \in [0, \frac{\pi}{6}]$, $\angle Op_2D = \angle Op_2C \in [\frac{\pi}{4}, \frac{7\pi}{12}]$. The shapes of uncovered zones of Φ, which are HJF and GIE (the darker gray zones in Fig.6(a) and Fig.6(b)), remain unchanged in this interval. By calculating the areas of HJF, GIE and Φ, we reach the conclusion immediately. $\square$

In the following design, we use region Φ' as the approximate tracking region.

2) Stimulus Management Policy: Similar to LPSS with circle-based model, we let only the awakened sensors detecting the object during $[cT-\epsilon, cT)$ produce the sensing stimulus. However, to identify sensors inside Φ', the accumulation of received stimulus in the previous round, e.g. the $(c-1)$th round, will be kept as historical information till the end of current round. We name the sensing stimulus received during the previous round by *negative stimulus* while that during the current round by *positive stimulus*. The former is produced by awakened sensors inside $C(p_1, R_s)$, while the latter is produced by that inside $C(p_2, R_s)$ (Fig.5(a)). Given the communication radius R_c, a sensor with location p' can receive the negative and the positive stimulus produced by only the awakened nodes inside $C(p', R_c) \cap C(p_1, R_s)$ and $C(p', R_c) \cap C(p_2, R_s)$ respectively. The condition of $R_c \leqslant 2R_s$ guarantees that if $p' \in C(p_2, R_c)$ (the boundary is excluded), $C(p', R_c) \cap C(p_1, R_s) \neq \emptyset$ and $C(p', R_c) \cap C(p_2, R_s) \neq \emptyset$. Thus each sensor v inside Φ' can hear both the negative and positive sensing stimulus because $\Phi' \subseteq C(p_2, R_c)$. Each sensor accumulates and stores the received negative stimulus as I^- and the received positive stimulus as I^+.

3) State Transition Rule: Next we design the state transition rule to identify nodes inside region Φ'. Because the object moves with constant velocity v, $\|p_1p_2\| = vT$.

Let Ω be the region of $\Phi' - C(p_2, R_s)$, e.g. the gray zone in Fig.5(b). We first consider the nodes inside $C(p_2, R_s)$, and then the nodes inside Ω. For region $C(p_2, R_s)$, since all sensors detected the object in the previous round locate in $C(p_2, R_s)$, we simply let them keep awake in the current round to serve detection.

For nodes inside Ω, the situation becomes complicated and we need the help of received stimulus I^- and I^+. Consider a sensor v with location p' inside Φ'. Let $S^+_{p'}$ be the area of $C(p', R_c) \cap C(p_2, R_s)$, and $S^-_{p'}$ be the area of $C(p', R_c) \cap C(p_1, R_s)$. Given the node density ρ and the wake-up probability P_t, obviously I^+ and I^- of v depend on $S^+_{p'}$ and $S^-_{p'}$. As demonstrated in Fig.5(b), let $\angle p'p_2O = \beta$. Before giving a theorem about both $S^-_{p'}$ and $S^+_{p'}$, we present two lemmas.

Lemma 1. *For each sensor with location $p' \in \Omega$,*

$$S^-_{p'} = \mathcal{A}(R_s, R_c, (\|p_1p_2\|^2 + (\mathcal{A}^{-1})^2(R_s, R_c, S^+_{p'}) + 2\|p_1p_2\|\mathcal{A}^{-1}(R_s, R_c, S^+_{p'})\cos\beta)^{\frac{1}{2}}). \quad (1)$$

Proof. Because $p' \in \Omega$, both $\|p'p_1\|$ and $\|p'p_2\|$ are in $(R_c - R_s, R_c + R_s)$. According to Fact 1, $S^+_{p'} = \mathcal{A}(R_s, R_c, \|p'p_2\|)$ and $S^-_{p'} = \mathcal{A}(R_s, R_c, \|p'p_1\|)$. Theorem 1 guarantees that if the values of R_s and R_c are fixed, the inverse function of $\mathcal{A}$ exists. Hence

$$\|p'p_2\| = \mathcal{A}^{-1}(R_s, R_c, S^+_{p'}). \quad (2)$$

Since $S^-_{p'} = \mathcal{A}(R_s, R_c, \|p'p_1\|)$, applying the Cosine theorem and combining Eq.(2), this lemma holds. □

Lemma 2. *For each sensor with location $p' \in \Omega$, if the value of $S^+_{p'}$ is fixed, $S^-_{p'}$ monotonically increases with β when $\beta \in [0, \pi]$.*

Proof. When $0 \leqslant \beta \leqslant \pi$, $\cos\beta$ monotonically decreases with β. According to Theorem 1 and Eq.(1), when the value of $S^+_{p'}$ is fixed, $S^-_{p'}$ monotonically increases with β. □

Let $\mathcal{F}(x, \beta) = \mathcal{A}(R_s, R_c, (\|p_1p_2\|^2 + (\mathcal{A}^{-1})^2(R_s, R_c, x) + 2\|p_1p_2\|\mathcal{A}^{-1}(R_s, R_c, x)\cos\beta)^{\frac{1}{2}})$, at last we have:

Theorem 4. *When $\sigma \in [0, \frac{\pi}{6}]$, for each sensor with location p', $p' \in \Omega$ iff (1) $\pi R_s^2 > S^+_{p'} \geqslant \mathcal{A}(R_s, R_c, R_c)$, and (2) $S^-_{p'} \leqslant \mathcal{F}(S^+_{p'}, 2\sigma + \frac{\pi}{4})$.*

Proof. Because $\sigma \in [0, \frac{\pi}{6}]$, for p' inside Ω, it is apparent that both $\|p'p_1\|$ and $\|p'p_2\|$ are in $(R_c - R_s, R_c + R_s)$.

$\Rightarrow$: Because $\Omega \subset C(p_2, R_c)$, when $p' \in \Omega$, according to the proof of Theorem 2, (1) holds. According to Eq.(1), $S^-_{p'} = \mathcal{F}(S^+_{p'}, \angle p'p_2O)$. Noting that $\angle p'p_2O \leqslant C'p_2O = 2\sigma + \frac{\pi}{4}$, according to Lemma 2, (2) holds.

$\Leftarrow$: When $\pi R_s^2 > S^+_{p'} \geqslant \mathcal{A}(R_s, R_c, R_c)$, according to the proof of Theorem.2, $p' \in C(p_2, R_c) - C(p_2, R_s)$. In the case of $S^-_{p'} \leqslant \mathcal{F}(S^+_{p'}, 2\sigma + \frac{\pi}{4})$, according to Lemma 2, $\angle p'p_2O \leqslant 2\sigma + \frac{\pi}{4}$, hence $p' \in \Omega$ holds. □

We propose the state transition rules based on the above results next. For nodes inside $C(p_2, R_s)$, we simply let the sensors detected the object in the previous round keep awake. For nodes inside Ω we need to consider both I^+ and I^-. Since the expected value of I^+ is $\rho\Delta_I P_t S^+_{p'}$, given I^+, $S^+_{p'}$ can be approximated as $\frac{I^+}{\rho\Delta_I P_t}$. Let N_1 be the number of awakened nodes inside zones with area $\mathcal{F}(\frac{I^+}{\rho\Delta_I P_t}, 2\sigma + \frac{\pi}{4})$.

For each sensor:
<u>*In the sensing phase, e.g. $((c-1)T, cT - \tau]$:*</u>
1. Compute $[\widehat{N}, +\infty)$ with confidence P_1;
2. Compute $(-\infty, \widehat{N_1}]$ with confidence P_2;
3. If detected object in the previous round, remain awake; else if $I^+ \geqslant \widehat{N}\Delta_I$ and $I^- \leqslant \widehat{N_1}\Delta_I$, enter the tracking state, otherwise enter the detecting state;
4. Let $I^- = I^+$ and Clear I^+.
<u>*In the communication phase, e.g. $(cT - \tau, cT]$:*</u>
<u>Concurrently execute the following two steps:</u>
1. Receive sensing stimulus and store in I^+;
2. If detected the object during $[cT - \epsilon, cT)$: If it is in the detecting state then produce sensing stimulus $\Delta_{I'}$, otherwise produce sensing stimulus Δ_I.

Fig. 7. LPSS with Probability-based Model.

For a sensor u with location $p' \in \Omega$, Theorem 4 indicates that: (1) the number of awakened nodes inside $C(p_2, R_s) \cap C(p', R_c)$ is supposed to be greater than or equal to N because $S^+_{p'} \geqslant \mathcal{A}(R_s, R_c, R_c)$ (N is defined as in the last subsection), and (2) the number of awakened nodes inside $C(p_1, R_s) \cap C(p', R_c)$ is supposed to be less than or equal to N_1 because $S^-_{p'} \leqslant \mathcal{F}(S^+_{p'}, 2\sigma + \frac{\pi}{4}) \approx \mathcal{F}(\frac{I^+}{\rho\Delta_I P_t}, 2\sigma + \frac{\pi}{4})$. Thus $N\Delta_I$ is supposed to a lower bound of I^+, and $N_1\Delta_I$ is supposed to be a upper bound of I^-. Accordingly we propose the tracking condition as $I^+ \geqslant N\Delta_I$ *and* $I^- \leqslant N_1\Delta_I$, and the state transition rule is designed as follows.

Rule 2. *For each sensor inside Ω, if $I^+ \geqslant N\Delta_I$ and $I^- \leqslant N_1\Delta_I$, enter the tracking state in the next round.*

Also, N_1 is a random variable and thus cannot be computed accurately. As N_1 serves as a upper bound, it can be estimated with one-side confidence interval $(-\infty, \widehat{N_1}]$. The details of interval estimation and the corresponding confidences P_2 will be discussed in Section VI. We substitute $\widehat{N}$, $\widehat{N_1}$ for N, N_1 in Rule 2. Combining the stimulus management policy and the rules above, we propose LPSS with probability-based model as Fig.7.

VI. Practical Implementation Issues

In this section we address two important practical implementation issues: the interval estimation for N, N_1, and the calculation of $\mathcal{F}$ in LPSS with probability-based model.

A. Interval Estimation

Here we present the details of interval estimation for N, N_1, which are used in Rule 1 and 2. Given area S, node density ρ and wake-up probability P_t of each node, the number of awakened nodes X is a random variable following the Binomial distribution, i.e. $Pr[X = m] = \binom{n}{m} P_t^m (1 - P_t)^{n-m}$ where $n = \lfloor \rho S \rfloor$.

Therefore, by setting n to $\lfloor \rho\mathcal{A}(R_s, R_c, R_c) \rfloor$, given confidence P_1, the confidence interval $[\widehat{N}, +\infty)$ for N can be obtained by solving $\sum_{i=\widehat{N}}^{n} \binom{n}{i} P_t^i (1 - P_t)^{n-i} = P_1$. Similarly, by setting n to $\lfloor \rho\mathcal{F}(\frac{I^+}{\rho\Delta_I P_t}, 2\sigma + \frac{\pi}{4}) \rfloor$, given confidence P_2,

the confidence interval $(-\infty, \widehat{N_1}]$ for N_1 can be obtained by solving $\sum_{i=0}^{\widehat{N_1}} \binom{n}{i} P_t^i (1-P_t)^{n-i} = P_2$.

Below we discuss the effect of above interval estimation on the state transition rules. Let $\Phi_C = C(p_2, R_c)$ (Fig.3(b)), by substituting $\widehat{N}$ (with confidence P_1) for N in Rule 1, we have the following result.

Theorem 5. *For each sensor with location p' and amount of accumulated stimulus I,*

$$Pr[p' \in \Phi_C | I \geqslant \widehat{N}\Delta_I] > P_1.$$

Proof. For the convenience of description, we let $m = \widehat{N}$ and $d = \|p'p_2\|$.

$$\begin{aligned} Pr[p' \in \Phi_C | I \geqslant \widehat{N}\Delta_I] &= \frac{Pr[p' \in \Phi_C, I \geqslant m\Delta_I]}{Pr[I \geqslant m\Delta_I]} \\ &> Pr[p' \in \Phi_C, I \geqslant m\Delta_I] \\ &= \int_0^{R_c} Pr[d=l] Pr[I \geqslant m\Delta_I | d=l] \mathrm{d}l, \end{aligned}$$

$Pr[I \geqslant m\Delta_I | d=l] = \sum_{i=m}^{n} \binom{n}{i} P_t^i (1-P_t)^{n-i}$, where $n = \lfloor \mathcal{A}(R_s, R_c, l)\rho \rfloor$. $Pr[d=l] = \frac{1}{R_c}$. According to the properties of Binomial distribution and Theorem 1, $Pr[I \geqslant m\Delta_I | d=l]$ monotonically decreases with l. Hence,

$$\begin{aligned} &\int_0^{R_c} Pr[d=l] Pr[I \geqslant m\Delta_I | d=l] \mathrm{d}l \\ >& \frac{1}{R_c} \int_0^{R_c} Pr[I \geqslant m\Delta_I | d=R_c] \mathrm{d}l \\ =& Pr[I \geqslant m\Delta_I | d=R_c] = Pr[I \geqslant \widehat{N}\Delta_I] = P_1. \quad \square \end{aligned}$$

Theorem 5 shows that Rule 1 can identify nodes inside Φ_C with confidence greater than P_1. Let $\Phi_P = \Omega$ (Fig.5(b)), by substituting $\widehat{N_1}$ (with confidence P_2) for N_1 in Rule 2, we have the following result.

Theorem 6. *For each sensor with location p' and amount of accumulated negative stimulus I, given $\mathcal{A}(R_s, R_c, R_c) \leqslant S_{p'}^+ < \pi R_s^2$,*

$$Pr[p' \in \Phi_P | I \leqslant \widehat{N_1}\Delta_I] > P_2.$$

Proof. The proof is similar to that of Theorem 5 and is omitted for limited space. $\square$

Theorem 6 shows that Rule 2 can identify nodes inside Φ_P with confidence greater than P_2.

The confidences P_1 and P_2 directly affect $\widehat{N}$ and $\widehat{N_1}$, the lower or upper bounds in the state transition rules, and thus the number of sensors in the tracking state. Higher confidences will make extra nodes enter the tracking state and waste energy. In contrast, lower confidences will decrease the number of sensors in the tracking state, causing unexpectedly low level sensing stimulus. As a result, less and less sensors will enter the tracking state and the object will be lost. This phenomenon is called a run with failure. The reasonable settings of P_1 and P_2 will be discussed through experiments in Section VII-A3.

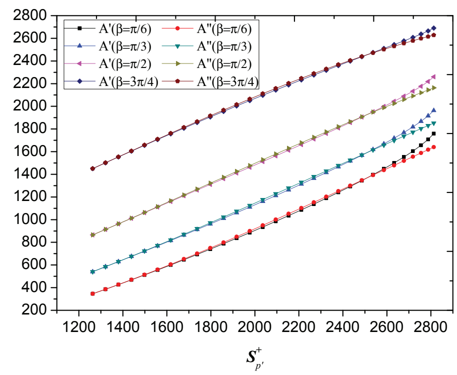


Fig. 8. Example results of $\widehat{\mathcal{F}}$, where A', A'' are the exact value and approximate result of $S_{p'}^-$ respectively. ($d(p', p_2) \in [30, 60]$)

B. Approximation of $\mathcal{F}$

The function $\mathcal{F}$ is used in the estimation of confidence interval $(-\infty, \widehat{N_1}]$. However, the calculation of $\mathcal{F}$ is impractical since it is difficult to obtain the inverse function $\mathcal{A}^{-1}$. For simplifying computation, we find an approximation function $\widehat{\mathcal{F}}$ for $\mathcal{F}$ as below.

Fixing R_s and R_c, we find a linear fitting function $\widehat{\mathcal{A}}(d)$ for $\mathcal{A}(R_s, R_c, d)$ first. Applying Taylor expansion at $d = \frac{(R_c+R_s)+(R_c-R_s)}{2} = R_c$, we have:

$$\widehat{\mathcal{A}}(d) = \mathcal{A}'(R_s, R_c, R_c)(d - R_c) + \mathcal{A}(R_s, R_c, R_c),$$

where $\mathcal{A}'(R_s, R_c, R_c) = -\frac{\sqrt{-R_s^4 + 4R_s^2R_c^2}}{R_c}$. Hence we obtain an approximation of $\mathcal{A}^{-1}$ as follows:

$$\widehat{\mathcal{A}^{-1}}(R_s, R_c, S) = \frac{1}{\mathcal{A}'(R_s, R_c, R_c)} S - \frac{\mathcal{A}(R_s, R_c, R_c)}{\mathcal{A}'(R_s, R_c, R_c)} + R_c. \tag{3}$$

Plug Eq.(3) into Eq.(1), we obtain $\widehat{\mathcal{F}}$ as follows:

$$\begin{aligned} \widehat{\mathcal{F}}(S_{p'}^+, \beta) \triangleq & \mathcal{A}(R_s, R_c, (\|p_1p_2\|^2 + (\widehat{\mathcal{A}^{-1}})^2(R_s, R_c, S_{p'}^+) \\ & + 2\|p_1p_2\| \widehat{\mathcal{A}^{-1}}(R_s, R_c, S_{p'}^+) \cos\beta)^{\frac{1}{2}}). \end{aligned}$$

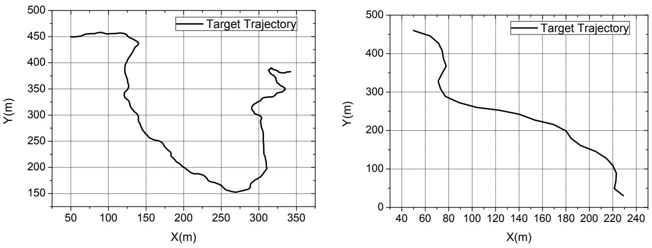
Examples in Fig.8 show that $\mathcal{F}$ is well approximated by $\widehat{\mathcal{F}}$ except a slight deviation when $S_{p'}^+ > 2600$.

VII. Performance Evaluation

In this section we evaluate the performance of LPSS through extensive experiments. The experiments are conducted on a custom network simulator.

A. Experiment Settings

1) Network Parameters: In simulations, we uniformly deploy 3000 nodes into a $500m \times 500m$ zone, and the node density is about 1 node$/100m^2$. R_s and R_c are set to $30m$ and $60m$ respectively. The length of each round, T, is set to 1 sec, and the length of the communication phase τ is set to 20 mSec. On the MICAz mote, 20 mSec is sufficient to transmit more than 70 packets of 64 bits. It means that given no more than 70 nodes, if each broadcasts one packet of 64 bits, the whole process can finish successfully using TDMA-like



(a) A random trajectory (b) A Gaussian trajectory

Fig. 9. Examples of target trajectories.

schemes. The value of ϵ is also set to 20 mSec, which causes a negligible location error of only $0.06m$ with the following settings of objects' motion patterns.

2) Motion Patterns of Object: The movement of objects with the random motion pattern and the Gaussian motion pattern are implemented. The maximum velocity of object is $30m$/sec. In the Gaussian motion pattern, σ is set to $\{\frac{\pi}{6}, \frac{\pi}{24}\}$. Examples of objects' trajectories with these two motion patterns are shown in Fig.9.

3) Determine P_1 and P_2: We define a metric, *success ratio*, as the ratio of the number of runs without failure to the total number of runs. According to the analysis in Section VI-B, low values of confidences P_1 and P_2 providing high success ratio are desirable. Based on the above network settings and objects' motion patterns, we carried out experiments and counted the number of successful runs of LPSS with different values of P_1 and P_2, i.e. from 0.05 to 1.0 with a step of 0.05, each with 50 runs. According to statistical results, when $P_1 \geqslant 0.55$, the success ratio of LPSS with circle-based model is higher than 80%. With $P_1 = 0.55$ and $P_2 \geqslant 0.85$, the success ratio of LPSS with probability-based model is higher than 75% when $\sigma \leqslant \frac{\pi}{6}$. Accordingly we set $P_1 = 0.55$ and $P_2 = 0.85$ in the following experiments.

4) Compared Protocols: The PW scheme in [2], though is not specifically designed for energy conserving, is employed as a compared protocol since it is considered the only existing location-free protocol. To make the comparison fair, we add the maximum velocity information of object to PW, reducing the size of tracking region from the circle with radius $R_s + 2R_c$ to $R_s + R_c$, which saves more energy. Also, the prediction-based sleep scheduling protocol with accurate location information is implemented as the ideal baseline.

5) Observed Metrics: Two important metrics, namely the *number of average awakened nodes* and the *coverage probability*, are observed in experiments by sampling the network state during the tracking process. The number of average awakened nodes specifying the energy cost during the tracking process, is computed by averaging the number of awakened nodes in tracking regions in all samples. The coverage probability specifying the tracking quality, is defined as the percentage of samples in which the required coverage level k is satisfied. For each figure, all data are averages of 10 runs with different network deployments and object's moving trajectories. In each

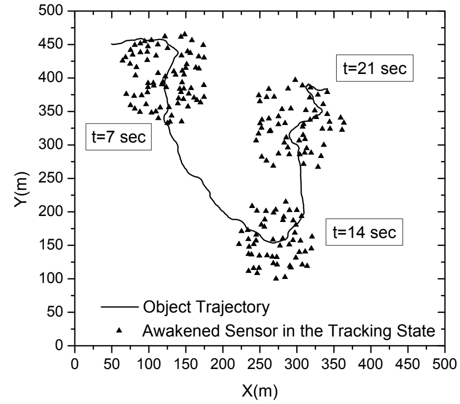


Fig. 10. Snapshots of awakened sensors in the tracking state in LPSS-Circle.

run, object moves for 25 sec and we sample the network state every 20 mSec.

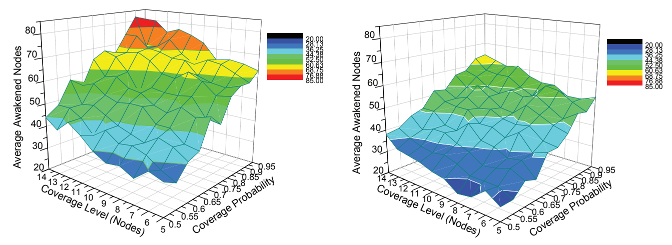
B. Experiment Results

We conduct evaluations and comparisons of LPSS in scenarios that objects move with random motion pattern first, and then in scenarios that objects move with Gaussian motion pattern.

1) Scenarios with Random Motion Pattern: In this kind of scenarios, without the probabilistic knowledge of objects, LPSS with probability-based model cannot be used. Therefore only the PW, LPSS with circle-based model (LPSS-Circle) and ideal prediction-based sleep scheduling for circle-based model (Ideal-Circle) are implemented and compared.

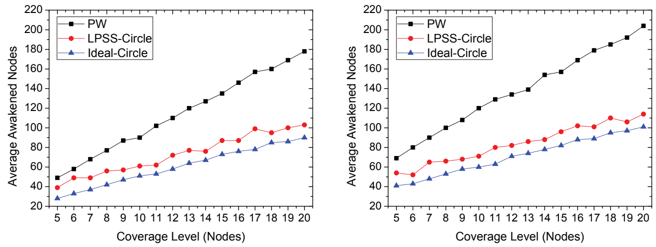
As shown in Fig.10, in LPSS-Circle, the awakened sensors in the tracking state approximately spread in circular regions as expected. We then examine the effects of required coverage level k and probability P_{req} on the number of average awakened nodes in LPSS-Circle. As shown in Fig.11(a), in LPSS-Circle, the number of average awakened nodes becomes larger as k and P_{req} increase. That is because when k and P_{req} increase, the wake-up probability of each node in tracking region increases and thus leads to more awakened nodes.

To present the results of the comparisons among these three protocols, we set P_{req} to 0.75 and 0.95, which are relative lower and higher values respectively. According to Fig.12, the



(a) LPSS-Circle (b) LPSS-Prob ($\sigma = \frac{\pi}{6}$)

Fig. 11. The number of average awakened nodes with required coverage level and probability.



(a) $P_{req} = 0.75$ (b) $P_{req} = 0.95$

Fig. 12. The number of average awakened nodes in scenarios with random motion pattern.

average awakened nodes in LPSS-Circle is much less than that in PW. As k increases, compared to PW, LPSS-Circle saves at most 40% awakened nodes. That is because the circular tracking region in PW is with radius $R_s + R_c$, resulting in a larger area than that in LPSS-Circle. We also see that the number of average awakened nodes in LPSS-Circle is very close to that in Ideal-Circle. The slight disparity comes from the setting of P_1, which causes more than desired awakened nodes but guarantees a high success ratio.

To see the actual coverage probability provided by LPSS-Circle, we set the required coverage probability P_{req} from 0.5% to 0.95% with a step of 0.05%. With each value of P_{req}, we compute the average value and the deviation of actual coverage probabilities in cases with k =5 to 20. The results demonstrated in Fig.13 show that LPSS-Circle guarantees the coverage probability well.

2) Scenarios with Gaussian Motion Pattern: The PW, LPSS-Circle, LPSS with probability-based model (LPSS-Prob) and ideal prediction-based sleep scheduling for probability-based model (Ideal-Prob) are implemented and compared here. Without losing the generality, we set $\sigma = \frac{\pi}{6}$ and $\frac{\pi}{24}$ in LPSS-Prob respectively.

As shown in Fig.14, in LPSS-Prob, the awakened sensors in the tracking state approximately spread in circular sector-like regions as expected. We then examine the effects of k and P_{req} on the number of average awakened nodes in LPSS-Prob. As shown in Fig.11(b), similar to that in LPSS-Circle, the number of average awakened nodes becomes larger as k and P_{req} increase.

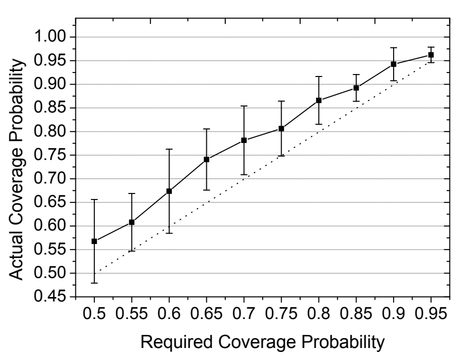


Fig. 13. Actual coverage probability in LPSS-Circle.

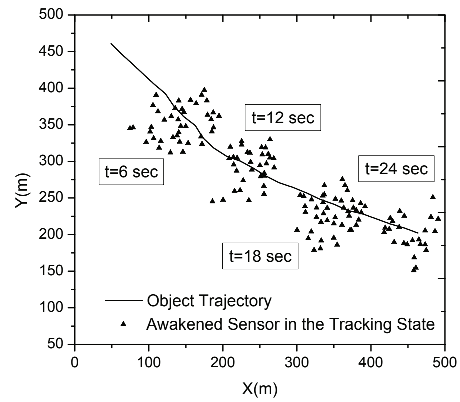
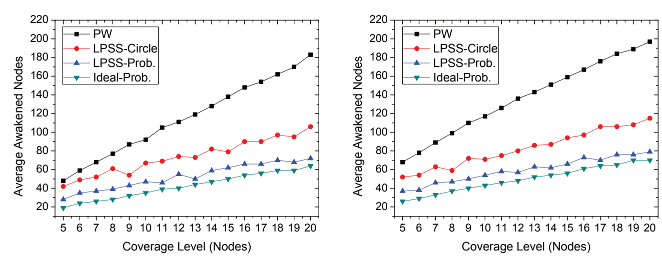
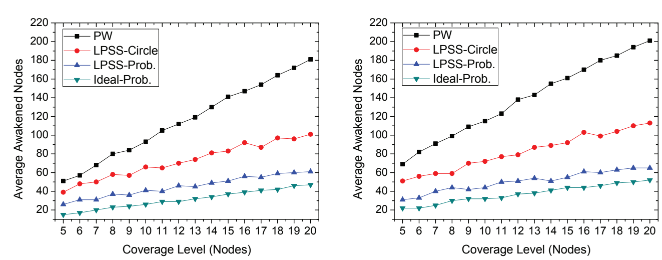


Fig. 14. Snapshots of awakened sensors in the tracking state in LPSS-Prob. ($\sigma = \frac{\pi}{24}$, $P_{req} = 0.85$ and $k = 10$)

To present the results of the comparisons among the four protocols, we also set P_{req} to 0.75 and 0.95. As shown in Fig.15, the average awakened nodes in LPSS-Prob is much less than in PW. As k increases, compared to PW, LPSS-Prob saves at most 60% awakened nodes when $\sigma = \frac{\pi}{6}$ (Fig.15(a) & 15(b)), and at most 70% awakened nodes when $\sigma = \frac{\pi}{24}$ (Fig.15(c) & 15(d)). That is because the circular tracking region in PW is with radius $R_s + R_c$, whose area is much larger than that in LPSS-Prob, which is approximately a circular sector. We also see that LPSS-Prob saves about 30% to 40% of awakened nodes of LPSS-Circle. This is because the circular sector-like tracking region in LPSS-Prob is smaller than the circular tracking region in LPSS-Circle, which can be seen intuitively by comparing Fig.10 and Fig.14. As shown in



(a) $P_{req} = 0.75, \sigma = \frac{\pi}{6}$ (b) $P_{req} = 0.95, \sigma = \frac{\pi}{6}$



(c) $P_{req} = 0.75, \sigma = \frac{\pi}{24}$ (d) $P_{req} = 0.95, \sigma = \frac{\pi}{24}$

Fig. 15. The number of average awakened nodes in scenarios with Gaussian motion pattern.

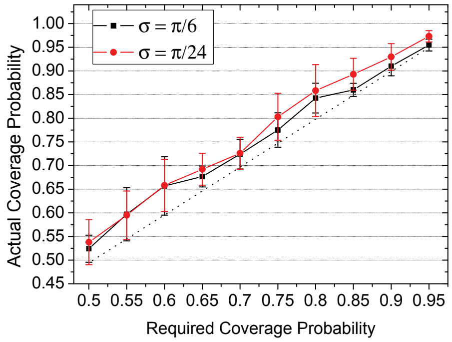


Fig. 16. Actual coverage probability in LPSS-Prob.

Fig.15, LPSS-Prob saves more awakened nodes with smaller σ. Moreover, we see that the number of average awakened nodes in LPSS-Prob is very close to that in Ideal-Prob. The slight disparity comes from the settings of P_1 and P_2, which causes more than desired awakened nodes but guarantees a high success ratio.

We examine the actual coverage of LPSS-Prob using the same methodology in the experiment evaluating the actual coverage probability of LPSS-Circle. According to Fig.16, we see that LPSS-Prob also guarantees the coverage probability well.

VIII. Conclusions

Saving energy while maintaining satisfactory tracking qualities, sleep scheduling protocols for object tracking attracted a lot of attention recently. Such approaches are called the prediction-based sleep scheduling protocols. However, most existing prediction-based sleep scheduling protocols require sensor nodes to know the locations of themselves, which may not always be available. In this paper we proposed a Location-free Prediction-based Sleep Scheduling protocol (LPSS) for object tracking in sensor networks. LPSS scheduled the wake-up and sleep of sensors according to the production and the reaction of sensing stimulus, without requiring location information. Thus LPSS can be used in a wider range of application environments. LPSS guaranteed the coverage level, an important tracking quality in most applications. We implemented LPSS with two most popular prediction models: the Circle-based and the Probability-based prediction models. Experiment results showed that LPSS not only provided a qualified coverage level, but also saved about 40% to 70% energy compared to existing location-free protocols. Moreover, the energy cost of LPSS is close to the ideal approach using accurate location information in terms of the number of awakened nodes. The design of LPSS with more prediction models and the implementation of a prototype system will be the future research.

Acknowledgment

This work is partially supported by the Hong Kong Research Grant Council under CERG grant PolyU 5102/07E, the Hong Kong Polytechnic University under the Niche Area grant 1-BB6C, NSFC (90718031, 60721002, 60803111) and 973 Program (2006CB303000, 2009CB320705).

References

[1] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. A. Stankovic, and T. Abdelzaher, “Achieving real-time target tracking using wireless sensor networks,” in *RTAS ’06*, 2006.

[2] C. Gui and P. Mohapatra, “Power conservation and quality of surveillance in target tracking sensor networks,” in *MobiCom ’04*, 2004.

[3] W. Wang, V. Srinivasan, K.-C. Chua, and B. Wang, “Energy-efficient coverage for target detection in wireless sensor networks,” in *IPSN ’07*, 2007.

[4] S. Pattem, S. Poduri, and B. Krishnamachari, “Energy-quality tradeoffs for target tracking in wireless sensor networks,” in *IPSN ’03*, 2003.

[5] Y. Xu, J. Winter, and W.-C. Lee, “Prediction-based strategies for energy saving in object tracking sensor networks,” in *MDM ’04*, 2004.

[6] X. Wang, J. J. Ma, S. Wang, and D. W. Bi, “Prediction-based dynamic energy management in wireless sensor networks,” *Sensors*, vol. 7, pp. 251–266, 2007.

[7] J. Jeong, T. Hwang, T. He, and D. Du, “Mcta: Target tracking algorithm based on minimal contour in wireless sensor networks,” in *INFOCOM ’07*, 2007.

[8] J. Fuemmeler and V. Veeravalli, “Smart sleeping policies for energy efficient tracking in sensor networks,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 2091–2101, May 2008.

[9] B. Jiang, K. Han, B. Ravindran, and H. Cho, “Energy efficient sleep scheduling based on moving directions in target tracking sensor network,” in *IPDPS ’08*, 2008.

[10] V. Seshadri, G. Zaruba, and M. Huber, “A bayesian sampling approach to in-door localization of wireless devices using received signal strength indication,” in *PerCom ’05*, 2005.

[11] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, “Tracking a moving object with a binary sensor network,” in *SenSys ’03*, 2003.

[12] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, “On target tracking with binary proximity sensors,” in *IPSN ’05*, 2005.

[13] N. Shrivastava, R. M. U. Madhow, and S. Suri, “Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms,” in *SenSys ’06*, 2006.

[14] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, “Tracking multiple targets using binary proximity sensors,” in *IPSN ’07*, 2007.

[15] Y. Busnel, L. Querzoni, R. Baldoni, M. Bertier, and A.-M. Kermarrec, “On the deterministic tracking of moving objects with a binary sensor network,” in *DCOSS ’08*, 2008.

[16] F. Zhao, J. Shin, and J. Reich, “Information-driven dynamic sensor collaboration,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, Mar 2002.

[17] S. Aeron, V. Saligrama, and D. Castaon, “Efficient sensor management policies for distributed target tracking in multihop sensor networks,” *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2562–2574, June 2008.

[18] R. Tan, G. Xing, J. Wang, and H. C. So, “Collaborative target detection in wireless sensor networks with reactive mobility,” in *IWQoS ’08*, 2008.

[19] C. fan Hsin and M. Liu, “Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms,” in *IPSN ’04*, 2004.

[20] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, “Towards optimal sleep scheduling in sensor networks for rare-event detection,” in *IPSN ’05*, 2005.

[21] A. Keshavarzian, H. Lee, and L. Venkatraman, “Wakeup scheduling in wireless sensor networks,” in *MobiHoc ’06*, 2006.

[22] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang, “Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 3, pp. 334–350, March 2007.

[23] Y. Wu, S. Fahmy, and N. Shroff, “Energy efficient sleep/wake scheduling for multi-hop sensor networks: Non-convexity and approximation algorithm,” in *INFOCOM ’07*, 2007.

[24] K. Premkumar and A. Kumar, “Optimal sleep-wake scheduling for quickest intrusion detection using wireless sensor networks,” in *INFOCOM ’08*, 2008.

[25] M. Hefeeda and H. Ahmadi, “A probabilistic coverage protocol for wireless sensor networks,” in *ICNP 2007*, 2007.